

- Chapitre de cours: [Schémas XML](#)
- Le [TD 5](#) sur les schémas (avec des exemples de schémas)
- `xmllint --noout --schema monschema.xsd monfichier.xml`

# 1

1. Donnez un document XML qui correspond au schéma suivant:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:simpleType name="IntOrName">
    <xsd:union memberTypes="xsd:integer xsd:Name"/>
  </xsd:simpleType>
  <xsd:complexType name="MonBeauTypeMixe" mixed="true">
    <xsd:sequence>
      <xsd:element name="data" type="IntOrName" nillable="true"
        minOccurs="3" maxOccurs="unbounded"/>
      <xsd:any processContents="skip"/>
    </xsd:sequence>
    <xsd:anyAttribute processContents="skip"/>
  </xsd:complexType>
  <xsd:element name="root" type="MonBeauTypeMixe"/>
</xsd:schema>
```

2. Si nécessaire, étoffez votre XML pour qu'il exerce véritablement tous les facettes du schéma donné en exemple:
  1. La mixité de contenu de `<root>`
  2. La dualité de types de `<data>`
  3. L'*élément arbitraire* peut contenir n'importe quoi. Faites-le contenir du contenu mixte!
  4. L'*attribut arbitraire* peut en fait être répété plusieurs fois. Essayez!
  5. (\*) L'élément `<data>` est `nillable`. Rendez-en un vide! (cf [Cours](#); on doit introduire une namespace)

3. Vérifiez que votre XML est bien étoffé, selon les points ci-dessus:

```
wget http://fabien.viger.free.fr/xml/td6/td6.1.tar.gz
tar -zxf td6.1.tar.gz
./td6.1.test.sh votre_exemple.xml
```

Corrigé: [XML](#)

## 2 xsd:key, xsd:keyref

1. Donner un schéma qui valide le XML suivant:

```
<library xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <book>
    <isbn>978-1-447-26455-2</isbn>
    <title>Gone with the wind</title>
    <language>EN</language>
    <pages>1024</pages>
    <published>1936-06-10</published>
    <author ref="mmitchell"/>
    <editor ref="warner"/>
  </book>
  <author id="mmitchell">
    <name>Margareth Mitchell</name>
    <book ref="978-1-447-26455-2"/>
  </author>
  <editor id="warner">
    <name>Warner Bros</name>
  </editor>
</library>
```

```
<book ref="978-1-447-26455-2"/>
```

```
</editor>
```

```
</library>
```

2. Utilisez `xsd:key` et `xsd:keyref` pour forcer les références à être cohérentes. Cf [Cours](#).
3. Vérifiez et/ou corrigez votre schéma pour qu'il soit toujours valide avec cet [exemple plus étoffé](#).
4. Vérifiez que votre schéma est valide et strict (par rapport aux références):

```
wget http://fabien.viger.free.fr/xml/td6/td6.2.tar.gz
tar -zxf td6.2.tar.gz
./td6.2.test.sh basic votre_schema.xsd
```

5. Rendez votre schéma aussi strict que possible; sur chaque champ (en restant logique sur le contenu accepté).

Vérifiez que vous êtes aussi strict que possible.

```
./td6.2.test.sh strict votre_schema.xsd
```

### 3

1. Donnez un exemple de XML validé par le schéma suivant:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://ma.namespace/perso"
  elementFormDefault="qualified">
  <xsd:element name="doc">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="hello">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="world"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Corrigé: [XML](#)

2. On complique un peu le schéma (en rouge, les zones modifiées ou ajoutées). Adaptez votre XML.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  targetNamespace="http://ma.namespace/perso"
  elementFormDefault="qualified">
  <xsd:import namespace="http://www.w3.org/1999/xhtml"
    schemaLocation="http://fabien.viger.free.fr/xml/td6/xhtml11-strict.xsd"/>
  <xsd:element name="doc">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="hello">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="world"/>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element ref="xhtml:p"/>
      </xsd:sequence>
    </xsd:complexType>
```

```
</xsd:element>
</xsd:schema>
```

*Note: On utilise une version "light" du schéma XHTML, normalement disponible sur <http://www.w3.org/2002/08/xhtml/xhtml1-strict.xsd>, car la version réelle ralentit xmllint considérablement.*

Corrigé: [XML](#)

### 3. On complique encore le schéma. Adaptez votre XML.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  targetNamespace="http://ma.namespace/perso"
  elementFormDefault="qualified">
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://fabien.viger.free.fr/xml/td6/xml.xsd"/>
  <xsd:import namespace="http://www.w3.org/1999/xhtml"
    schemaLocation="http://fabien.viger.free.fr/xml/td6/xhtml1-strict.xsd"/>
  <xsd:element name="doc">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="xhtml:html"/>
        <xsd:element name="unsorted_html_elements">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:any processContents="lax" minOccurs="3" maxOccurs="unbounded"
                namespace="http://www.w3.org/1999/xhtml"/>
            </xsd:sequence>
            <xsd:attribute ref="xml:id" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

Corrigé: [XML](#)

### 4. Pour finir, on va référencer les version officielles des schémas de la namespace

<http://www.w3.org/XML/1998/namespace> et de <http://www.w3.org/1999/xhtml>. Vous verrez que xmllint rame. Essayez d'étoffer un peu le HTML à l'intérieur de votre XML.

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  targetNamespace="http://ma.namespace/perso"
  elementFormDefault="qualified">
  <xsd:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/xml.xsd"/>
  <xsd:import namespace="http://www.w3.org/1999/xhtml"
    schemaLocation="http://www.w3.org/2002/08/xhtml/xhtml1-strict.xsd"/>
  <xsd:element name="doc">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element ref="xhtml:html"/>
        <xsd:element name="unsorted_html_elements">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:any processContents="lax" minOccurs="3" maxOccurs="unbounded"
                namespace="http://www.w3.org/1999/xhtml"/>
            </xsd:sequence>
            <xsd:attribute ref="xml:id" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

```
    </xsd:element>
</xsd:schema>
```